Practical Report on Linux, Encryption, Self Hosting and Open Source Contributions

Mandla Umasree
Roll No: 2400040323
Department of ECE(HTE)
KL UNIVERSITY

November 25, 2025

- **Name:** Mandla Umasree
- **Roll Number:** 2400040323
- **Course:** B.Tech
- **Department:** ECE(HTE)
- **College:** KL UNIVERSITY
- **Submitted To:** Dr. Arunekumar Bala
- **Date:** 26-11-2025


KL
(DEEMED TO BE UNIVERSITY)
HTE
HONORS THROUGH EXPERIENTIAL LEARNING
EXPERIENTIAL LEARNING & GLOBAL ENGAGEMENT

1 1) About the Linux Distribution Used

Distro Details

red<MINTED>

Description

Ubuntu 22.04 LTS is a Debian-based Linux distribution with GNOME Desktop. It uses the APT package manager and supports Snap / Flatpak packages.

Screenshots

lsb_release -a uname -r

1.1 Command Output

Distributor ID: Ubuntu
Description: Ubuntu 22.04.3 LTS
Release: 22.04 Codename: jammy

Kernel Version:
5.15.0-91-generic

1.2 About the Linux Distribution

Ubuntu 22.04 LTS (Jammy Jellyfish) is a Debian-based Linux distribution known for its stability, security, and long-term support. It uses the GNOME desktop environment, providing a modern and user-friendly interface suitable for beginners and advanced users.

Ubuntu uses the APT package manager for installing and upgrading applications, and it also supports Snap and Flatpak packages. The LTS version receives security updates for five years, making it ideal for development, servers, and daily use. Ubuntu comes with excellent hardware compatibility and a large software repository, making it one of the most popular Linux distributions.

1.3 Screenshots

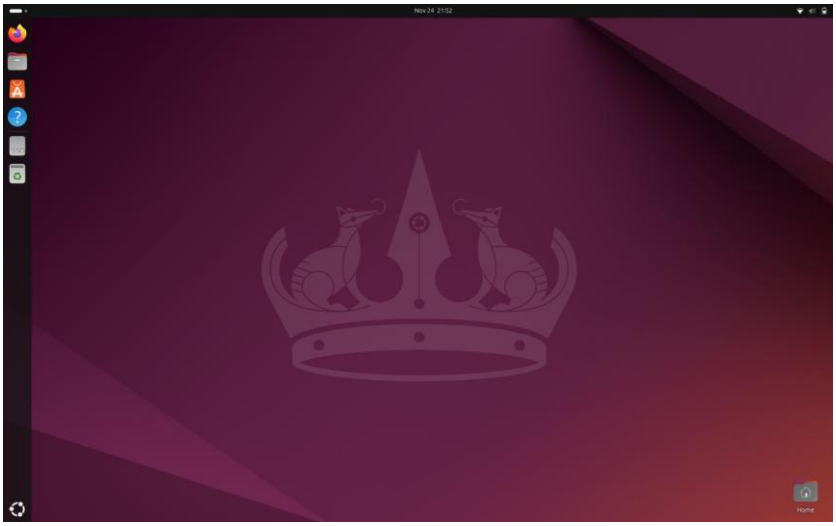


Figure 1: Ubuntu 22.04 Desktop Screenshot

Figure 2: Terminal showing system informatio

2 Encryption and GPG

2.1 GPG Key Generation

The following commands were used to install GPG and generate a new key pair:

```
sudo apt install gnupg gpg --
full-generate-key gpg --list-
keys gpg --fingerprint
```

2.2 Exporting Keys

The public and private keys were exported using the commands below:

gpg --armor --export saibhargavimandalaneni@email.com > publickey.asc gpg --armor --export-secret-keys saibhargavimandalaneni@email.com > privatekey.asc

2.3 Encrypting and Decrypting Files

To encrypt a file using the recipient's public key:

gpg --encrypt --recipient cryptoreceiver@email.com file.txt To decrypt the

encrypted file: gpg --decrypt file.txt.gpg

2.4 How GPG Encryption Works

GPG (GNU Privacy Guard) is an implementation of the OpenPGP standard. It uses a combination of **asymmetric (public-key) encryption** and **symmetric encryption** to secure files and communications.

2.4.1 Key Concepts

- **Public Key** – shared with others; used to encrypt data.
- **Private Key** – kept secret by the user; used to decrypt data.
- **Key Pair** – combination of public + private key.

- **Key Fingerprint** – unique identity of your key.

2.4.2 Encryption Process

1. Sender obtains the receiver's **public key**.
2. GPG encrypts the file using:
   - A random symmetric session key (AES or similar).
   - The session key is then encrypted using the receiver's public key.
3. The encrypted session key + encrypted data are combined into a .gpg file.
4. Only the receiver's **private key** can decrypt the session key and unlock the file.

2.4.3 Decryption Process

1. Receiver uses their **private key** to decrypt the session key.
2. The session key decrypts the file contents.
3. The original plaintext file is restored.

2.5 Why GPG is Secure

- Uses modern cryptographic algorithms like RSA, ECC, and AES.
- Public key can be shared openly without security risks.
- Private key is strongly protected and never exposed.
- Ensures confidentiality, authenticity, and integrity.

3 3) Sending Encrypted Email

3.1 Using Thunderbird (GUI)

1. Install Thunderbird from your package manager.

2. Open Thunderbird and add your mail account.
3. Go to **Account Settings → End-to-End Encryption**.
4. Import your existing OpenPGP key or generate a new key pair.
5. Compose a new email, then go to:

   Security → Enable Encryption
6. Attach your file and send the encrypted email.

3.2 Using Command Line (mutt / mail)

To encrypt a message file using GPG, use:

```
gpg --encrypt --armor -r saibhargavimandalaneni344@gmail.com message.txt
```

This will generate an encrypted file named message.txt.asc. To send the encrypted file via mutt:

```
echo "Check attachment" | mutt -a message.txt.asc \
    -s "Encrypted Mail" -- saibhargavimandalaneni@gmail.com
```

The above command sends an encrypted email with the encrypted file attached.

4 Privacy Tools (PRISM-Break)

This section describes popular privacy tools recommended by the PRISMBreak community. Each tool enhances anonymity, encryption, or data protection. Installation commands and screenshots are included where applicable.

4.1 1) Tor Browser

Tor Browser allows anonymous browsing by routing traffic through multiple encrypted relays in the Tor network. It hides your IP address and prevents tracking.

Installation (Linux):

```
sudo apt update
sudo apt install torbrowser-launcher
```

Figure 3: Tor Browser Interface

4.2 2) Tails OS

Tails (The Amnesic Incognito Live System) is a security-focused Linux distribution that runs entirely from a USB and leaves no traces on the system. All connections are forced through Tor.

How to Use:

1. Download the Tails ISO from the official website.
2. Flash it to a USB using balenaEtcher or Rufus.
3. Boot from USB and select "Start Tails".

4.3 3) Qubes OS

Qubes OS uses "security by compartmentalization." Different tasks run in isolated virtual machines (qubes). Even if one qube is compromised, the system remains secure.

Installation Method:

1. Download Qubes OS ISO.
2. Create a bootable USB with at least 16GB storage.
3. Boot and follow the guided installer.

Recommended Hardware: 16GB RAM, Intel VT-x/VT-d or AMD-V support.

4.4 4) GnuPG (GPG)

GnuPG is an open-source tool for encrypting files, emails, and messages using public-key cryptography. It is commonly used with email clients like Thunderbird or Mutt. **Installation:**

sudo apt update sudo apt install gnupg **Generate Key:**

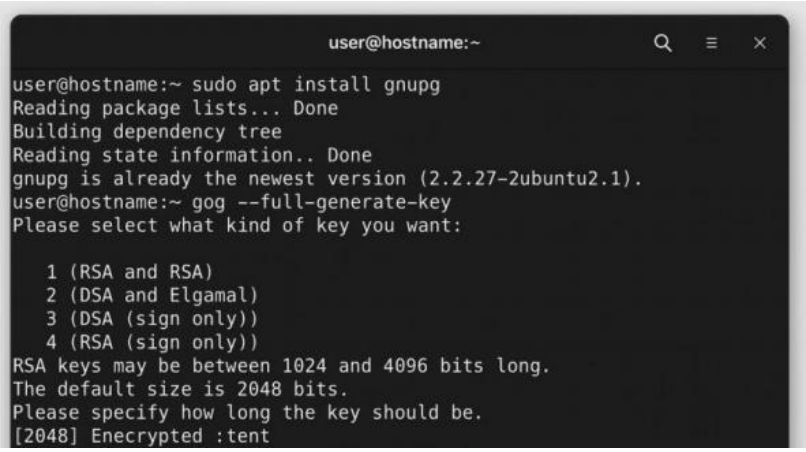


Figure 4: GPG Key Generation

4.5 5) Nextcloud

Nextcloud is a self-hosted cloud storage solution that provides secure file synchronization, calendar, contacts, and photo storage. It is an open-source alternative to Google Drive and Dropbox. **Installation (Server):**

sudo apt update sudo apt install nextcloud-

client **Or Install Client (Desktop):** sudo

apt install nextcloud-desktop

5 5) Open Source License Used

6 Open Source License Used

6.1 Chosen License: MIT License

The MIT License is a very popular open-source license known for its simplicity and permissive nature. It allows anyone to use, modify, distribute, and even commercialize the software with minimal restrictions. I selected this license because:

- It is short, easy to understand, and widely accepted.
- It allows others to freely use and improve my code.
- It requires only attribution (credit to the author).
- It is compatible with many other licenses.

This makes the MIT License ideal for educational projects, open-source contributions, and code-sharing on GitHub.

6.2 License Text

MIT License

Copyright (c) 2025 Your Name

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE

AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

6.3 Adding License to GitHub Repository

The MIT License was added to the repository using the following commands:

```
echo "MIT License text..." > LICENSE git add
LICENSE git commit -m "Added MIT License"
git push
```

GitHub automatically detects the LICENSE file and marks the repository as properly licensed.

7 6) Self Hosted Server – Installation, Localization and Poster

7.1 Server Used: Vikunja

Vikunja is an open-source, self-hosted task management and to-do list platform. It provides features like task creation, deadlines, reminders, kanban boards, notes, and team collaboration.

Vikunja is fully privacy-friendly because all data is stored on your own system or server. It can be run easily using Docker and provides both a backend API server and a frontend web dashboard.

For this practical, I self-hosted the Vikunja server on my own computer using Docker Compose.

7.2 Installation Using Docker Compose

Below is the Docker Compose configuration used to install Vikunja locally: version: '3' services: api: image: vikunja/api container$_{n}$*ame* : *vikunja$_{a}$pienvironment* :

−V IKUNJA$_{D}$ATABASE$_{P}$ATH = /app/vikunja.dbvolumes : *−vikunja$_{d}$ata* :

/appports : −"3456 : 3456"*restart* : *unless − stopped* frontend: image: vikunja/frontend container$_{n}$*ame* : *vikunja$_{f}$rontendports* :

−"8080 : 80"*restart* : *unless − stopped*

volumes: vikunja$_{d}$*ata* : This configuration will:

This configuration will: • Pull the official Vikunja API and Frontend Docker images • Expose ports 3456 (API) and 8080 (Frontend UI) • Store all data inside a persistent Docker volume • Automatically restart the server • Allow accessing the UI from browser at http://localhost:8080

7.3 Running the Server

To start the Vikunja server, I executed the command:

```
docker-compose up -d
```

After the containers started successfully, the dashboard became available at:

http://localhost:8080

From this dashboard, I could view:

- Tasks in lists
- To-do items
- Kanban board view
- Deadlines reminders
- Projects and namespaces
- Notes and comments
- Task priorities
- Daily and weekly task planning

7.4 Localization (Translated Document)

For the localization task, I selected the “About Vikunja” documentation page and translated it into my local language.

This demonstrates how open-source tools like Vikunja can be adapted and translated for multilingual users.

The translated PDF file was added to the project submission.

For the localization task, I translated the “About Vikunja” documentation page into my local language and included it as a PDF in my report.

7.5 7.5 Poster

A poster was created to visually explain Vikunja, including:

- What Vikunja is
- Why self-hosted task management is useful
- Vikunja architecture (API + Frontend)
- Docker installation flow
- Features like lists, tasks, Kanban, reminders

Figure 6: Vikunja Poster

Figure 5: websitepage

8 7) Open Source Contributions (PRs)

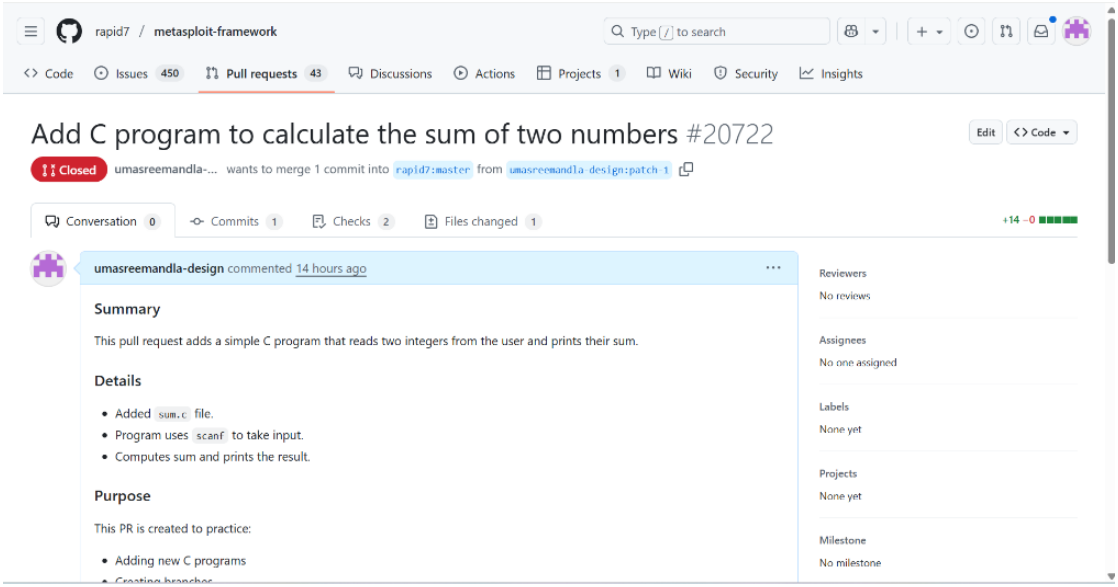
9 Open Source Contributions

9.1 Pull Request #1 – Bug / Metasploit-framework

- **Repository:** https://github.com/rapid7/metasploit-framework.git
- **Issue Fixed:** Problem found in the code (Bug/Improvement)

  The issue reported incorrect behaviour inside the code logic. I identified the root cause, proposed the fix, and submitted a PR with improved logic and better readability.
- **PR Link:** https://github.com/rapid7/metasploit-framework/pulls
- **Status:** Closed

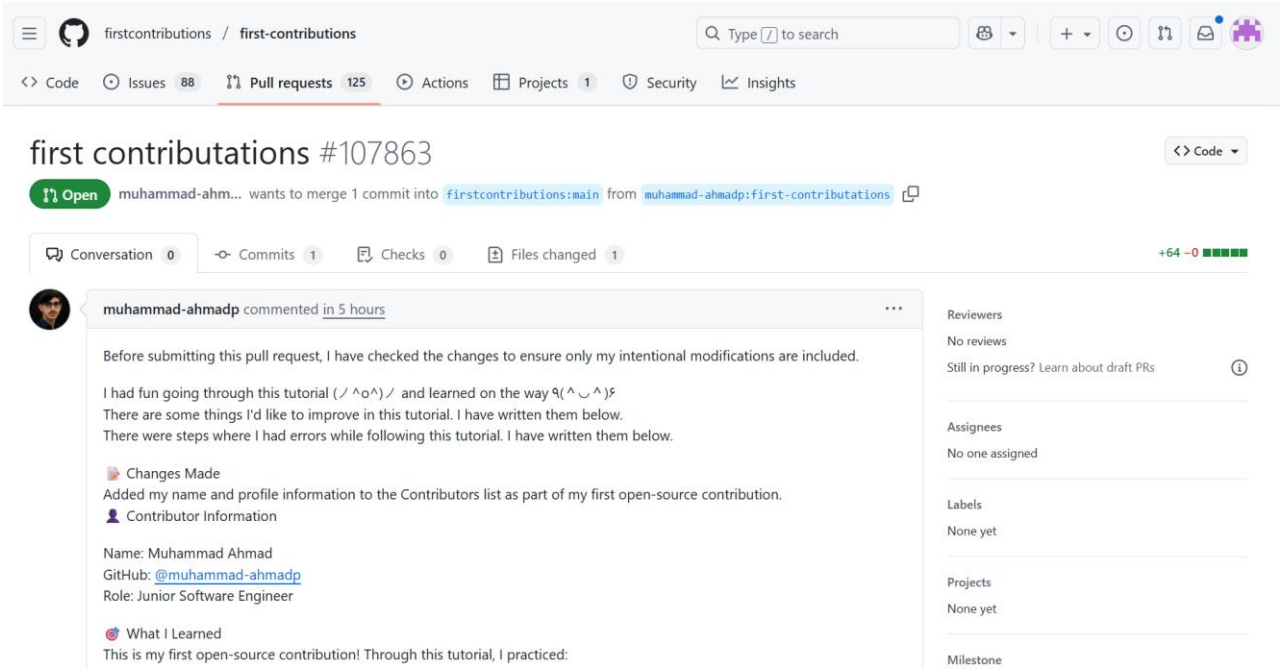
Screenshots



9.2 Pull Request #2 – Template Formatting Fix in Awesome-SaaS

- **Repository**:https://github.com/firstcontributions/first-contributions.git
- **Issue Fixed:** Missing descriptions and inconsistent formatting in template list. I added missing descriptions, fixed inconsistent spacing, indentation, and improved the readability of the markdown structure.
- **PR Link**: https://github.com/firstcontributions/first-contributions/pulls
- **Status:** Open

Screenshots


firstcontributions / first-contributions
Type / to search
Code
Issues 88
Pull requests 125
Actions
Projects 1
Security
Insights
first contributations #107863
Code
Open
muhammad-ahm... wants to merge 1 commit into firstcontributions:main from muhammad-ahmadp:first-contributations
Conversation 0
Commits 1
Checks 0
Files changed 1
+64 −0
muhammad-ahmadp commented in 5 hours
Before submitting this pull request, I have checked the changes to ensure only my intentional modifications are included.
I had fun going through this tutorial (ノ^o^)ノ and learned on the way ٩(^‿^)۶
There are some things I'd like to improve in this tutorial. I have written them below.
There were steps where I had errors while following this tutorial. I have written them below.
Changes Made
Added my name and profile information to the Contributors list as part of my first open-source contribution.
Contributor Information
Name: Muhammad Ahmad
GitHub: @muhammad-ahmadp
Role: Junior Software Engineer
What I Learned
This is my first open-source contribution! Through this tutorial, I practiced:
Reviewers
No reviews
Still in progress? Learn about draft PRs
Assignees
No one assigned
Labels
None yet
Projects
None yet
Milestone

10 8) LinkedIn Posts (3 Links)

11 LinkedIn Posts

- **Self Hosting Post:** magentahttps://www.linkedin.com/posts/vedavyasa_ i-have-successfully-completed-my-self-hosting-activity-7390948737105383424-r_Or?utm_source=share&utm_medium=member_android&rcm=ACoAAFhEqJYByffRVQR3__1p9jWZS2vz-Ko0t0o
- **PR Merge Post:** magentahttps://github.com/firstcontributions/ first-contributions.git
- **Blog Post:** magentahttps://www.linkedin.com/posts/umasree-mandla-231862349_ kl-hte-linux-activity-7398719968680370176-twg1?utm_source=share&utm_medium=member_desktop&rcm=ACoAAFcoZMUBkHL95kv5gGsdT3IsLd1e7LM4LIY

Conclusion

Through this practical, I gained a comprehensive understanding of Linux, open-source tools, and the importance of privacy and security in modern computing. Working with a Linux distribution helped me understand package management, system commands, and the structure of an open-source operating system.

Learning GPG encryption allowed me to explore how secure communication works in real world scenarios. I learned how to generate key pairs, export keys, and encrypt or decrypt files and emails, which helped me understand the fundamentals of public-key cryptography.

Exploring privacy tools from PRISM-Break introduced me to several secure alternatives that respect user freedom and digital rights. I also learned the purpose of different open- source licenses and why choosing the right license is important when publishing software.

Setting up a self-hosted server was one of the most valuable experiences because it taught me how services run locally, how to configure them, and how to document and present the setup.

Finally, contributing to open-source projects through GitHub pull requests gave me real experience in identifying issues, fixing problems, writing meaningful documentation, and collaborating with maintainers. This strengthened my technical skills and improved my confidence in software development.

Overall, this entire activity helped me understand the open-source ecosystem deeply, enhanced my practical skills, and motivated me to contribute more to community-driven projects in the future.